



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/612,724	06/30/2003	Kalyan Muthukumar	884.889US1	3799
21186	7590	09/11/2006	EXAMINER	
SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A. P.O. BOX 2938 MINNEAPOLIS, MN 55402				FRANCIS, MARK P
ART UNIT		PAPER NUMBER		
		2193		

DATE MAILED: 09/11/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/612,724	MUTHUKUMAR ET AL.	
	Examiner	Art Unit	
	Mark P. Francis	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 30 June 2003.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-38 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-11, 13-26, and 28-38 is/are rejected.
- 7) Claim(s) 12 and 27 is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on 30 June 2003 is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) Notice of References Cited (PTO-892)
- 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date 08/30/04.
- 4) Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.
- 5) Notice of Informal Patent Application
- 6) Other: _____.

DETAILED ACTION

1. This action is responsive to the application filed on June 30, 2003.
2. Claims 1-38 have been examined.

Oath/Declaration

3. The Office acknowledges receipt of a properly signed oath/declaration filed June 30, 2003.

Claim Rejections - 35 USC § 101

4. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

5. Claims 1-8 and 33-38 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

According to the 101 Interim Guidelines, The tangible requirement does not necessarily mean that a claim must either be tied to a particular machine or apparatus or must operate to change articles or materials to a different state or thing. However, the tangible requirement does require that the claim must recite more than a § 101 judicial exception, in that the process claim must set forth a practical application of that § 101 judicial exception to produce a real-world result. Benson, 409 U.S. at 71-72, 175 USPQ at 676-77 (invention ineligible because had "no substantial practical application."). "[An application of a law of nature or mathematical formula to a ... process may well be deserving of patent protection." Diehr, 450 U.S. at 187, 209 USPQ at 8 (emphasis added); see also 21 Corning, 56 U.S. (15 How.) at 268, 14 L.Ed. 683 ("It is for the

discovery or invention of some practical method or means of producing a beneficial result or effect, that a patent is granted . . ."). In other words, the opposite meaning of "tangible" is "abstract."

Regarding claims 1 and 5,

In this instance, the language of the claim raises a question as to whether the claim is directed merely to an abstract idea that is not tied to an environment or machine which would result in a practical application that would produce a useful, concrete, and tangible result to form the basis of statutory subject matter under 35 USC 101.

Applicant defines a method for parallelizing execution of loop computations that includes indirectly accessed sparse arrays that uses software pipelining to increase instruction-level parallelism without defining any of the steps that are needed to perform the parallelizing execution of loop computations. This renders claim 1 to be an abstract idea, which can be implemented using software means only. Thus, the claim as a whole does not result in a tangible result of a practical application.

Regarding Claim 33,

Applicant defines a method for software-pipelining a loop by dynamic run-time disambiguation of elements of an index array using idle issues slots in a software pipelining schedule. Applicant only defines software pipelining a loop by dynamic run-time disambiguation of the elements by using idle issues slots but does not disclose or

define a tangible result that can be implemented on some form of computer readable storage medium or hardware. Therefore, all of the steps inside the body of the claim form an abstract idea that can all be implemented using software means only. Therefore, all of the steps can all be implemented using software means only that do not produce a tangible result of a practical application.

Regarding claim 34,

Although Applicant defines in the preamble a computer readable medium that stores computer executable instructions, the body of the claim merely defines software pipelining a loop by dynamic run-time disambiguation of the elements by using idle issues slots but does not disclose or define a tangible end result that can be implemented on some form of computer readable storage medium. Therefore, all of the steps inside the body of the claim form an abstract idea that can all be implemented using software means only that do not produce a tangible result of a practical application which is not patentable under 35 U.S.C. 101.

The rejection of the base claim are incorporated into their dependent claims.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:
7. A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

8. Claims 1-38 are rejected under 35 U.S.C. 102(e) as being anticipated by Thompson. (U.S. PG Pub 2003/0237080)

Independent claims

With respect to claim 1, Thompson discloses a method comprising parallelizing execution of loop computations that involve indirectly accessed sparse arrays/matrices using software-pipelining to increase instruction-level parallelism(Col 2:0011, "...instruction level parallelism...") and decrease initiation interval. (Col 3:0021-0023, "...programming loop...minimum initiation interval...",e.g. See Fig. 6A element 610 and related text)

With respect to claim 5, Thompson discloses a method comprising: parallelizing execution of loop computations,(Col 5:0049-0052, "...associates an identifier with each program loop...") by transforming dependence loop instructions into multiple independent iterations, (Col 3:0021-0023, "...program dependence...") in sparse arrays/matrices using software-pipelining to break recurrence initiation interval such that recurrence initiation interval is substantially closer to resource initiation interval. (Col 3:0021-0023, "...programming loop...minimum initiation interval...",e.g. See Fig. 6A

element 610 and related text)

With respect to claim 9, Thompson discloses a method comprising: performing a run time dependency check during a current iteration using prior computed values obtained from a predetermined number of previous adjacent iterations in a sparse array matrix; (Col 3:0021-0023, "...program dependence...") and parallelizing dependent loop instructions between the current iteration and subsequent multiple iterations by using the computed values to make the dependence loop instructions into independent loop iterations so as to increase instruction-level parallelism(Col 3:0021-0023, "...program dependence...") and reduce recurrence initiation interval in the current iteration as a function of the run time dependency check during the current iteration. (Col 3:0021-0023, "...programming loop...minimum initiation interval...",e.g. See Fig. 6A element 610 and related text)

With respect to claim 15, Thompson discloses a method comprising: transforming sparse array matrix code to perform a run time dependency check using a predetermined number of prior computed values; (Col 3:0021-0023, "...program dependence...") software-pipelining(Col 2:0011, "...software pipelining...") the transformed sparse array matrix code to perform the run time dependency check in a current iteration using the predetermined number of prior computed values; (Col 2:0011, "...instruction level parallelism...")

and software-pipelining(Col 2:0011, "...software pipelining...") to parallelize multiple iterations by overlapping execution of dependence loop instructions in the prior computed values to reduce recurrence initiation interval in the sparse array matrix based on the run time dependency check. (Col 3:0021-0023, "...programming loop...minimum initiation interval...",e.g. See Fig. 6A element 610 and related text)

With respect to claim 20, Thompson discloses a method comprising: transforming loop computations from a predetermined number of prior adjacent iterations in sparse arrays/matrices to current loads code to perform a run time dependency check using a predetermined number of prior computed values; (Col 3:0021-0023, "...program dependence...")

software-pipelining(Col 2:0011, "...software pipelining...") the transformed loop computations from the predetermined number of prior adjacent iterations to perform a run time dependency check in a current iteration using the predetermined number of prior computed values; (Col 2:0011, "...software pipelining...") and parallelizing the loop computations using the prior computed values. (Col 2:0011, "...instruction level parallelism...")

With respect to claim 26, Thompson discloses an article comprising a computer-readable medium which stores computer-executable instructions,(Col 4:0039-0041, "...and nonvolatile-memory elements...") the instructions causing a computer to: performing a run time dependency check during a current iteration using prior computed

values obtained from a predetermined number of previous adjacent iterations in a sparse array matrix; (Col 3:0021-0023, "...program dependence...") and parallelizing dependent dependence loop instructions between the current and subsequent multiple iterations by using the computed values to make the dependence loop instructions into independent loop iterations. (Col 3:0021-0023, "...program dependence...")

With respect to claim 30, Thompson discloses a system comprising: a bus; (e.g. See Fig , element 112)

a processor coupled to the bus; (e.g. See Fig. 1 element 110)

a memory coupled to the processor;(e.g. See Fig. 1 element 120)

a network interface device; (e.g. See Fig. 1 element 114,116)

wherein execution of loop computations in indirectly accessed sparse arrays/matrices by the processor uses software-pipelining to increase instruction-level parallelism(Col 2:0011, "...software pipelining...") and decrease initiation interval(Col 3:0021-0023, "...programming loop...minimum initiation interval...",e.g. See Fig. 6A element 610 and related text) by performing:

transforming loop computations from a predetermined number of prior adjacent iterations in sparse arrays/matrices to current loads code to perform a run time dependency check using a predetermined number of prior computed values; (Col 3:0021-0023, "...program dependence...")

software-pipelining the transformed loop computations from the predetermined number of prior adjacent iterations to perform a run time dependency check in a current iteration using the predetermined number of prior computed values; (Col 2:0011, "...software pipelining...")

and parallelizing the loop computations using the prior computed values to reduce recurrence initiation interval in the undisambiguated pointer stores from the predetermined number of prior adjacent iterations to current loads code based on the run time dependency check. (Col 2:0011, "...software pipelining...", Col 3:0021-0023, "...program dependence...")

With respect to claim 33, Thompson discloses a method comprising software-pipelining a loop by dynamic run time disambiguation of elements of an index array using otherwise-idle issues slots in a software-pipelining schedule. (Col 2:0011, "...software pipelining...", Col 3:0021-0023, "...a modulo scheduler...")

With respect to claim 34, Thompson discloses an article comprising a computer-readable medium that stores computer-executable instructions, the instructions causing a computer to software-pipeline a loop by dynamic run time disambiguation of elements of an index array using otherwise-idle issues slots in a software-pipelining schedule. (Col 2:0011, "...software pipelining...", Col 3:0021-0023, "...a modulo scheduler...")

Dependent claims

With respect to claims 2 , the rejection of claim 1 is incorporated and further, Thompson discloses that software-pipelining comprises: parallelizing dependent loop body instructions between subsequent independent iterations.(Col 3:0021-0023, "...for the given loop based on program dependence...")

With respect to claim 3, the rejection of claim 2 are incorporated and further, Thompson discloses the decreasing initiation interval comprises: increasing resource initiation interval, wherein resource initiation interval is based on resource usage of the dependence loop and available processor resources and recurrence initiation interval is based on the instructions in the dependence loop body and latencies of the processor. (Col 3:0021-0023, "...for the given loop based on program dependence...", e.g. See Fig. 6A element 610 and related text)

With respect to claim 4, the rejection of claim 2 is incorporated and further, Thompson discloses that the dependent loop body instructions are based on clock cycles. (Col 6:0051, "..in clock cycles...")

With respect to claim 6, the rejection of claim 5 is incorporated and further, Thompson discloses using the software-pipelining to break the recurrence initiation interval comprises implementing loop-body instructions in parallel in the multiple independent iterations. (Col 3:0021-0023, "...for the given loop based on program dependence...",

e.g. See Fig. 6A element 610 and related text)

With respect to claim 7, the rejection of claim 6 is incorporated and further, Thompson discloses that the loop-body instructions comprise loop-body cycles. (Col 6:0056, "...for rotating register usage within the loop...")

With respect to claim 8, the rejection of claim 7 is incorporated and further, Thompson discloses that the resource initiation interval is based on resource usage of the loop and available processor resources and the recurrence initiation interval is based on the cycles in the dependence loop and latencies of the processor. (Col 3:0021-0023, "...for the given loop based on program dependence...", e.g. See Fig. 6A element 610 and related text)

With respect to claim 10, the rejection of claim 9 is incorporated and further, Thompson discloses that the prior computed values are based on virtual unrolling using a virtual unrolling factor. (Col 3:0021-0023, "...for the given loop based on program dependence...", e.g. See Fig. 6A element 610 and related text)

With respect to claim 11, the rejection of claim 10 is incorporated and further, Thompson discloses that the virtual unrolling factor includes three previous iterations. (Col 3:0021-0023, "...for the given loop based on program dependence...", e.g. See Fig. 6A element 610 and related text)

With respect to claim 13, the rejection of claim 9 is incorporated and further, Thompson discloses assigning the prior computed values to a predetermined number of adjacent registers. (Col 6:0053-0055, "...to assign static or global variables to rotating registers...")

With respect to claim 14, the rejection of claim 12 is incorporated and further, Thompson discloses further comprising: performing register rotation to include computed values obtained from the current iteration; and repeating performing the run time dependency check and parallelizing the dependence loop instructions to increase the instruction-level parallelism and reduce the recurrence initiation interval in a next iteration. (Col 3:0021-0023, "...for the given loop based on program dependence...", e.g. See Fig. 6A element 610 and related text)

With respect to claim 16, the rejection of claim 15 is incorporated and further, Thompson discloses that software-pipelining the transformed sparse array matrix code comprises: computing forming a predetermined number of variables based on a virtual unrolling factor; initializing the computed predetermined number of variables; loading the prior computed values into the predetermined number of variables; assigning the prior computed values to a predetermined number of substantially adjacent registers; and software-pipelining using the assigned prior computed values. (Col 3:0021-0023, "...for the given loop based on program dependence...", e.g. See Fig. 6A element 610

and related text)

With respect to claim 17, the rejection of claim 16 is incorporated and further, Thompson discloses further comprising: performing register rotation to include computed values obtained from the current iteration; and repeating the software-pipelining and using the register rotated computed values to reduce the recurrence initiation interval in a next iteration. (Col 6:0053-0055, "...to assign static or global variables to rotating registers...")

With respect to claim 18, the rejection of claim 17 is incorporated and further, Thompson discloses that the computed values obtained from the predetermined number of prior adjacent iterations are based on virtual unrolling by using a virtual unrolling factor. (Col 6:0053-0055, "...to assign static or global variables to rotating registers...")

With respect to claim 19, the rejection of claim 18 is incorporated and further, Thompson discloses that the virtual unrolling factor is three. (Col 6:0053-0055, "...to assign static or global variables to rotating registers...")

With respect to claim 21, the rejection of claim 20 is incorporated and further, Thompson discloses that the predetermined number of prior computed values is obtained from the predetermined number of prior adjacent iterations based on a virtual

unrolling factor. (Col 3:0021-0023, "...for the given loop based on program dependence...", e.g. See Fig. 6A element 610 and related text)

With respect to claim 22, the rejection of claim 21 is incorporated and further, Thompson discloses that the virtual unrolling factor is three. (Col 6:0053-0055, "...to assign static or global variables to rotating registers...")

With respect to claim 23, the rejection of claim 20 is incorporated and further, Thompson discloses the parallelizing the computations using the prior computed values comprise: overlapping execution of dependence loop instructions in multiple dependent iterations in the sparse arrays/matrices using the software-pipelining. (Col 2:0011, "...software pipelining...", Col 3:0021-0023, "...a modulo scheduler...")

With respect to claim 24, the rejection of claim 20 is incorporated and further, Thompson discloses further comprising: assigning the prior computed values to a predetermined number of adjacent register. (Col 6:0053-0055, "...to assign static or global variables to rotating registers...")

With respect to claim 25, the rejection of claim 24 is incorporated and further, Thompson discloses further comprising: performing register rotation to include computed values obtained from the current iteration. (Col 6:0053-0056 to rotating registers...")

With respect to claim 28, the rejection of claim 26 is incorporated and further, Thompson discloses that the instructions further cause the computer to assign the prior computed values to a predetermined number of adjacent register. (Col 6:0053-0055, "...to assign static or global variables to rotating registers...")

With respect to claim 29, the rejection of claim 28 is incorporated and further, Thompson discloses that the instructions further cause the computer to perform register rotation to include computed values obtained from the current iteration. (Col 6:0053-0055, "...to assign static or global variables to rotating registers...")

With respect to claim 31, the rejection of claim 30 is incorporated and further, Thompson discloses that the processor further assigns the prior computed values to a predetermined number of adjacent register. (Col 6:0053-0055, "...to assign static or global variables to rotating registers...")

With respect to claim 32, the rejection of claim 31 is incorporated and further, Thompson discloses that the processor further performs register rotation to include computed values obtained from the current iteration. (Col 6:0053-0055, "...to assign static or global variables to rotating registers...")

With respect to claim 35, the rejection of claim 33 is incorporated and further,

Thompson discloses that the dynamic run time disambiguation of the elements comprises: holding a plurality of prior computed index variables associated with the index array in a corresponding plurality of rotating registers. (Col 3:0021-0023, "...for the given loop based on program dependence...", e.g. See Fig. 6A element 610 and related text)

With respect to claim 36, the rejection of claim 33 is incorporated and further, Thompson discloses that the dynamic run time disambiguation of the elements comprises: comparing a currently-loaded value of an index variable associated with the index array to a prior computed value of the index variable. (Col 6:0053-0055, "...to assign static or global variables to rotating registers...")

With respect to claim 37, the rejection of claim 34 is incorporated and further, Thompson discloses a computer-readable medium which stores the computer-executable instructions of claim 34, wherein the instructions further cause the computer to use one of a plurality of prior computed index variables associated with the index array in place of a currently-loaded value of an index variable associated with the index array. (Col 6:0053-0055, "...to assign static or global variables to rotating registers...")

With respect to claim 38, the rejection of claim 34 is incorporated and further, Thompson discloses 38. The article comprising a computer-readable medium which stores the computer-executable instructions of claim 34, wherein the instructions further

cause the computer to use a currently-loaded value of an index variable associated with the index array after determining that the currently-loaded value of the index variable is not equal to one of a plurality of prior computed index variables associated with the index array. (Col 6:0053-0055, "...to assign static or global variables to rotating registers...")

Allowable Subject Matter

9. Claims 12 and 27 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Conclusion

10. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mark P. Francis whose telephone number is (571) 272-7956. The examiner can normally be reached on Mon-Fri 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571) 272-3719. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mark P. Francis
Patent Examiner
Art Unit 2193

Xena - Cr.
KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100